
SPiC-Aufgabe #5: Ampel

(15 Punkte, Abgabe bis Mittwoch, 06.07.2011, um 8:00, keine Gruppen)

Entwerfen Sie eine Steuerung für die Ampelanlage an einem Fußgängerüberweg in einer Datei `ampel.c`. Hierbei wird die den Autos zugewandte Ampel durch die LEDs `RED0`, `YELLOW0` und `GREEN0` dargestellt, die Fußgängerampel durch die LEDs `RED1` und `GREEN1` (kein gelbes Licht). Durch das Drücken von `BUTTON0` können Fußgänger eine Umschaltung anfordern. Die LED `BLUE1` signalisiert den Fußgängern, dass eine Umschaltanforderung entgegengenommen wurde.

Die Steuerung soll im Detail wie folgt arbeiten:

- Die Grünphase für die Autos dauert mindestens zehn Sekunden.
- Eine Umschaltanforderung wird durch Druck auf `BUTTON0` ausgelöst. Der Druck der Taste wird durch Aktivierung der LED `BLUE1` (Signal kommt) bestätigt. Diese LED wird beim Start der Umschaltung wieder deaktiviert. Tastendrucke werden bis zum Ende der Grünphase für Fußgänger ignoriert (nicht aktiv).
- Die Ampel schaltet nach fünf Sekunden oder nach Ablauf der minimalen Grünphase für die Autos um – je nachdem, welches Ereignis später eintrifft.
- Bei der Umschaltung wird die Autoampel zunächst auf Gelb (500 ms), dann Rot geschaltet. Nach einer Verzögerung von 2 s schaltet die Fußgängerampel dann auf Grün.
- Die Grünphase der Fußgängerampel soll exakt 5 s andauern. Die Fußgänger haben anschließend 2 s Zeit um die Fahrbahn zu räumen. Anschließend wechselt die Autofahrerampel auf Gelb-Rot (500 ms) und wieder auf Grün.
- Auf der Siebensegmentanzeige soll die verbleibende Rot-Dauer in Sekunden angezeigt werden (0 entspricht der Umschaltung auf Grün). Die Anzeige soll sowohl für Fußgänger als auch für Autofahrer verwendet werden. Sind beide Ampeln rot, bleibt die Anzeige der bisherigen Rotphase zugeordnet. Wird die Siebensegmentanzeige nicht mehr benötigt (0), soll diese abgeschaltet werden.
- In Ruhephasen soll der Mikrocontroller in einen Sleep-Modus wechseln (kein aktives Warten). Der Taster `BUTTON0` soll von Ihnen so konfiguriert werden, dass ein externer Interrupt den Mikrocontroller aus dem Sleep-Modus aufwecken kann.
- Verwenden Sie die Module `LED` und `7seg` der `libspicboard`, sowie die Funktionen `sb_timer_setAlarm` und `sb_timer_cancelAlarm` des `Timer`-Moduls (nicht `sb_timer_delay`). Die Verwendung des `Button`-Moduls ist nicht zulässig.

Hinweise:

- Das `7seg`-Modul verwendet das `Alarm`-Modul, welches Interrupts verwendet.
- Die Callbackfunktion des `Timer` modules wird aus dem Interrupt-Kontext aufgerufen.
- Unnötigen Berechnungen nach dem Aufwachen aus dem Sleep-Modus sollen vermieden werden.
- Achten Sie auf die korrekte Verwendung des `volatile`-Schlüsselworts. Beschreiben Sie in einem Kommentar zu jeder verwendeten `volatile`-Variable, weshalb Sie dieses Schlüsselwort dort benötigen.